

Simulando Funções de Alta Ordem via o Padrão de Projetos Template

Fernando Magno Quintão Pereira

27 de outubro de 2010

Questão 1

Imagine que você trabalhe com mecanismos de busca na web. Seu problema básico é receber uma palavra, e uma lista de páginas web, e mapear esta lista para uma outra, de valores booleanos. Esta segunda lista contém `true` nas posições que correspondem a páginas que contém a dita palavra, e `false` nas posições das páginas onde a palavra não ocorre.

- ▶ Como resolver este problema? Qual é o algoritmo básico?
- ▶ Será que é possível resolvê-lo em paralelo?

Questão 2

O que o problema da questão anterior tem a ver com os seguintes problemas:

- ▶ Construir a lista de aniversário dos alunos da sala.
- ▶ Atualizar o salário de todos os funcionários da companhia, de modo que eles recebam um aumento de 10%.
- ▶ Encontrar o número de estudantes matriculados em cada uma das disciplinas oferecidas pela UFMG.

Questão 3

Imagine um algoritmo de criptografia muito simples, que apenas substitui cada caractere do texto por um outro, de acordo com uma tabela.

- ▶ Alguém poderia dar um exemplo de uso deste algoritmo?
- ▶ Será que este é um algoritmo seguro?
- ▶ Como quebrar este algoritmo?

Questão 4

Para encontrar a chave do algoritmo de criptografia da questão anterior, podemos usar uma tabela de frequência de palavras.

- ▶ Projete um programa que receba como entrada uma lista das palavras presentes em um texto em Português, e construa um histograma das letras, isto é, uma tabela que mapeia cada letra para o número de vezes em que esta letra ocorre no texto.
- ▶ Quão paralelo é este problema?

Questão 5

O que o problema da questão anterior tem a ver com os problemas abaixo relacionados?

- ▶ Encontrar o aluno mais velho entre todos os alunos da sala?
- ▶ Encontrar o número total de letras em todas as palavras do dicionário de Português?
- ▶ Descobrir o total gasto por uma empresa com o pagamento dos salários de cada funcionário?

Questão 6

Algumas linguagens de programação possuem o conceito de funções de alta ordem. Estas funções permitem a criação de *templates*.

- ▶ O que são funções de alta-ordem?
- ▶ O que são templates?
- ▶ Quais linguagens possuem funções de alta-ordem?
- ▶ **Exemplos.**

Questão 7

Esta questão refere-se ao mapeamento de dados entre listas.

- ▶ Qual a forma geral do algoritmo?
- ▶ Quais os dados envolvidos?
- ▶ Quais as operações envolvidas?
- ▶ Represente as classes envolvidas via um diagrama de classes.
- ▶ Como se dará o reuso de código?

Questão 8

Implemente um mapa que mapeia uma lista de inteiros para outra lista de inteiros, onde cada elemento é incrementado em uma unidade.

- ▶ Como testar a sua aplicação?
- ▶ Quais são as condições de contorno?

Questão 9

Implemente um mapa que mapeia uma lista de Strings para uma lista de inteiros, onde cada elemento é o tamanho da string correspondente.

- ▶ Como testar a sua aplicação?
- ▶ Como apurar a complexidade de seu algoritmo?
- ▶ Qual o papel das classes anônimas nesta implementação?
- ▶ Qual é a sintaxe das classes anônimas?

Questão 10

Implemente um mapa que mapeia uma lista de Strings para uma lista de caracteres, onde cada elemento é o primeiro caracter da string correspondente.

- ▶ Como testar a sua aplicação?

Questão 11

Esta questão refere-se à redução de dados de uma lista.

- ▶ Qual a forma geral do algoritmo?
- ▶ Quais os dados envolvidos?
- ▶ Quais as operações envolvidas?
- ▶ Quais os tipos destas operações?
- ▶ Represente as classes envolvidas via um diagrama de classes.
- ▶ Como se dará o reuso de código?

Questão 12

Implemente um redutor que some os elementos de uma lista de inteiros.

- ▶ Como testar a sua aplicação?
- ▶ Quais são as situações de contorno?

Questão 13

Implemente um redutor que produza o histograma de letras usadas em uma lista de strings.

- ▶ Como testar a sua aplicação?
- ▶ Como medir o comportamento assintótico deste programa?

Questão 14

Templates podem ser usados em muitas situações diferentes.

- ▶ Como utilizar templates para aumentar o reuso de um algoritmo de ordenação?
- ▶ Como diferentes dados podem ser ordenados?
- ▶ Em que funções de comparação diferentes podemos pensar?

Questão 15

Utilize templates para que possamos implementar uma versão mais reutilizável do algoritmo de ordenação *quicksort*.

- ▶ Represente sua aplicação como um diagrama de classes UML.
- ▶ Como testá-la?
 - ▶ Que base de dados utilizar?
 - ▶ Quais as condições de contorno que devemos testar?
 - ▶ O que seriam bons testes para medir o desempenho da aplicação?
 - ▶ Como demonstrar, via um gráfico, o desempenho do algoritmo?
 - ▶ Que tipo de ruído pode comprometer a corretude de nosso gráfico?